VŠB - Technická univerzita Ostrava

Fakulta elektrotechniky a informatiky

BAKALÁŘSKÁ PRÁCE

2011 Martin Ileček

VŠB - Technická univerzita Ostrava

Fakulta elektrotechniky a informatiky

Katedra informatiky

Záznam aktivit uživatelů

Users Activity Log

2011 Martin Ileček

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Martin Ileček**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: Záznam aktivit uživatelů
Users Activity Log

Zásady pro vypracování:

Cílem práce je vytvořit software schopný zaznamenávat aplikace používané uživatelem a takto získaná data vyhodnotit - např. pro účely vykazování činností zákazníkům, apod.

Požadavky na software jsou zejména:
1. Záznam používaných aplikací při běhu na pozadí.
2. Komfortní uživatelské rozhraní pro prohlížení/prohledávání uložených záznamů.
3. Schopnost seskupovat záznamy.
4. Široké možnosti výstupních sestav.
5. Možnost přeposílání záznamů na jinou instanci aplikace.

Seznam doporučené odborné literatury:

Podle pokynů vedoucího bakalářské práce.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Mgr. Zdeněk Horák**

Datum zadání: 20.11.2009
Datum odevzdání: 06.05.2011

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

„Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.“

Dne ……. V Ostravě ……

Martin Ileček

PODĚKOVÁNÍ

Děkuji Mgr. Zdeňkovi Horákovi za hodnotné rady a odborné vedení během mé práce.

Abstrakt

Bakalářská práce se zabývá sledováním aktivit uživatelů využívající implementaci na platformě .NET Framework pomocí programovacího jazyku C#. V prvé části jsou uvedeny příklady již existujících řešení tohoto tématu a jejich srovnání. Následuje obecný popis Windows Service z .NET framework. Poté obecná specifikace, uživatelská příručka a implementace mého softwaru.

Cílem bakalářské práce je vytvoření aplikace, jež bude jednoduše, systematicky a efektivně mapovat aktivity uživatelů a předkládat jim tato data v přijatelné formě příjemného uživatelského rozhraní.

Klíčová slova monitorování aktivity uživatelů, služba Windows, mapování procesů

Abstract

This Bachelor thesis deals with the monitoring of user activity using an application built over the .NET Framework platform using the C# programming language. The first part provides examples of existing solutions to this problem and their comparison. The following part is a general description of the Windows Services technology through the perspective of the .NET framework. The last part contains software specifications, user guide and detailed description the implementation.

The aim of this work is to create an application with simple, systematic and efficient mapping of user activity. This application will present gathered data using comfortable and user-friendly interface.

Key Words monitoring activity of users, Windows service, process mapping

Seznam použitých symbolů a zkratek

XML – eXtensible Markup Language

WPF – Windows Presentation Foundation

HTML – HyperText Markup Language

URL – Uniform Resource Locator

ASP – Active Server Pages

Obsah

1. Úvod

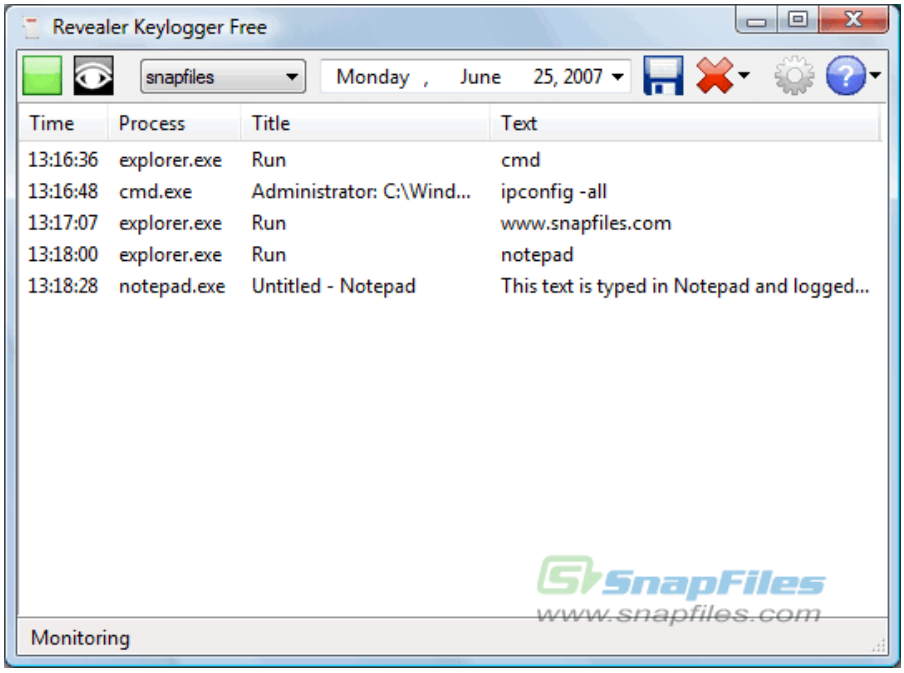
Toto téma bakalářské práce jsem si zvolil pro možnost implementovat nějaký reálný problém a také z důvodu, že výsledek této práce může být prakticky využit běžným uživatelem. Především v pracovním sektoru je dobré vědět, kolik času věnoval uživatel dané činnosti, a na základě těchto informací pak následně fakturovat svoji práci různým zákazníkům. I pro uživatele v domácnosti je mnohdy dobré vědět, čemu se věnují nejčastěji, a které procesy a programy mají nejčastěji a nejdéle puštěny. Z těchto informací pak mohou následně činit důležitá rozhodnutí například: zda zakoupit novou verzi softwaru, protože je ve firmě často využíván, či zda naopak používání některého softwaru nezakázat. Výsledky této aplikace mohou být přínosné i pro vývojáře. Mohou se dozvědět, které programy jsou nejčastěji používány současně, a tak mohou pracovat na jejich lepší koordinaci a propojení.

Toto řešení je určeno uživatelům samotným pro mapování jejich aktivit na počítači a má sloužit k zpětnému zhodnocení práce a využívání různých druhů software. Toto řešení není vytvořeno za účelem sledování aktivit uživatelů bez jejich vědomí a neslouží pro kontrolu třetích osob.

2. Již existující řešení

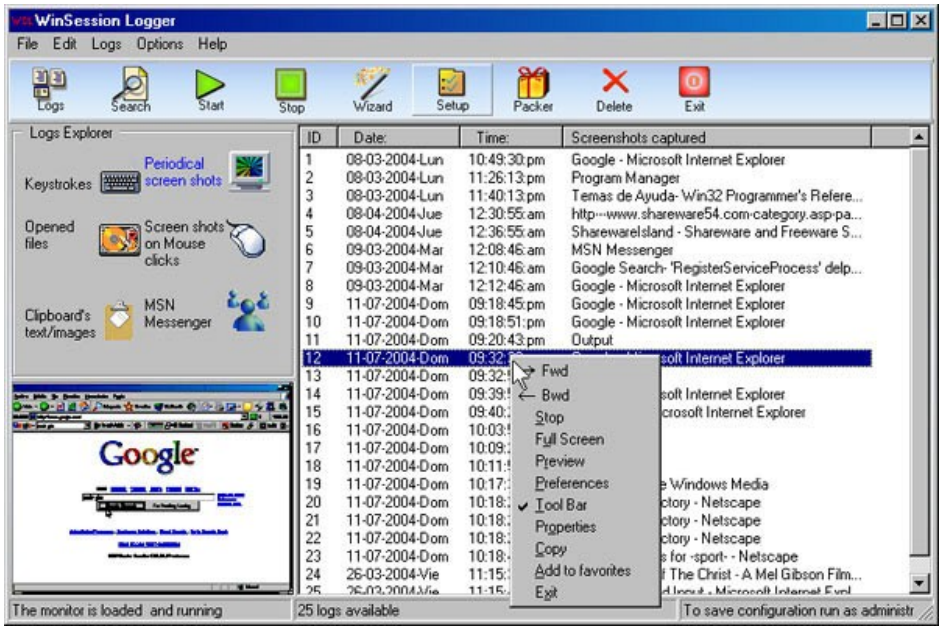
2.1.1. Revealer Keylogger (Free Edition)

Tento program umožňuje monitorování procesů, ukládaní tohoto výpisu do TXT souboru a má možnost být chráněn heslem. Může být nastaven na automatické spouštění po startu bez viditelného rozhraní.[1]



2.1.2. User Logger (Free Edition)

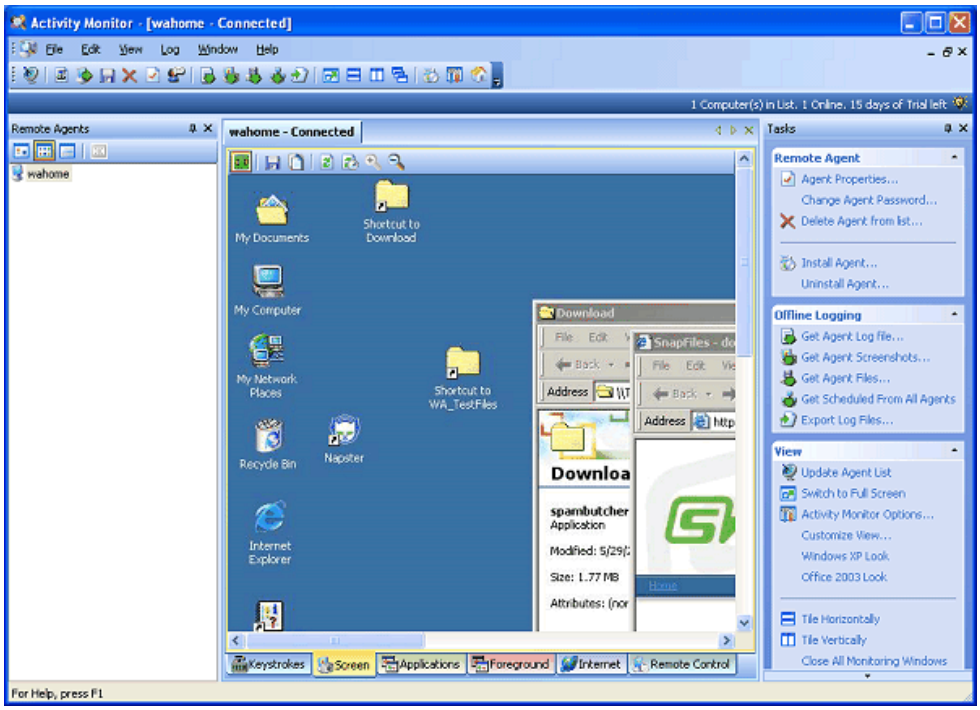
Jedná se o program monitorující probíhající procesy. Dokáže rozpoznat, jaký uživatel je přihlášen k počítači. Zaznamenává i vyfocené uživatelské rozhraní daného procesu. V plné verzi má tento program další rozšíření a funkce, například: šifrování, plánování, a další.[2]



2.1.3. Activity Monitor remote LAN surveillance

Tento program se používá pro vzdálenou kontrolu aktivit na daném počítači v LAN síti. Jedná se o profesionální řešení a placený produkt. Program umožňuje zobrazit probíhající aktivitu na daném počítači, a to i zpětně.

Umožnuje zobrazovat si spuštěné programy, otisky obrazovky, navštívené webové stránky, nebo které klávesy byla stisknuty. Tento program v sobě obsahuje možnost ovládání vzdáleného počítače: zapínání a vypínání programů, restartování a vypínání počítače, posílání zpráv na daný počítač pro komunikaci s uživatelem a další funkce.[3]

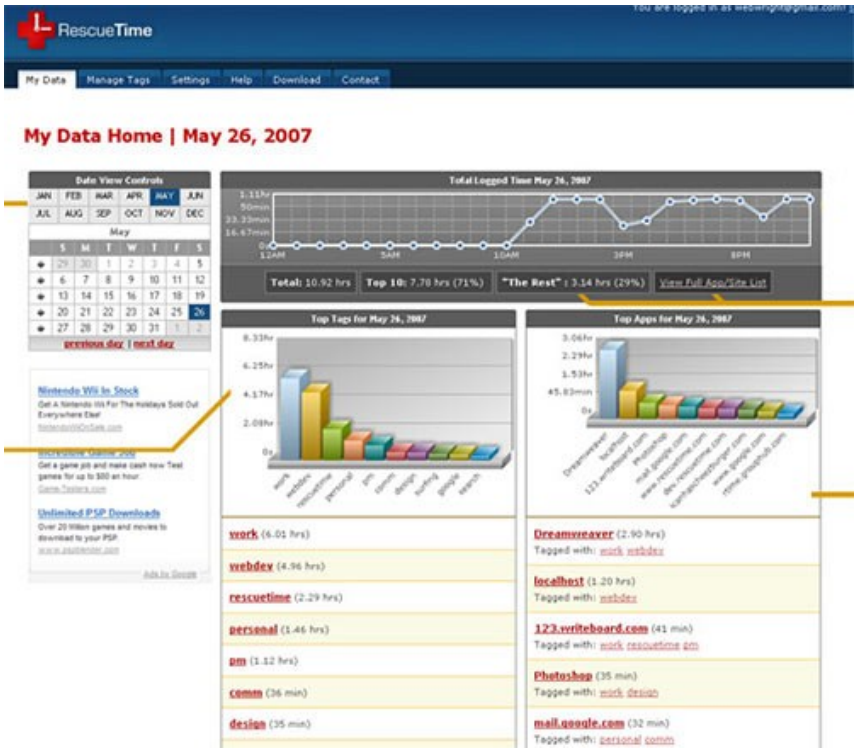


2.1.4.Spytech SpyAgent

Tento sledovací program je již profesionálním řešením a jedná se o placený produkt. Obsahuje však velké množství funkcí a možností pro sledování aktivit uživatelů. Mezi tyto možnosti patří především: zaznamenávání stisknutých kláves, otevřená okna, běžící aplikace, navštívené internetové stránky, odeslané a přijaté e-maily i otisky obrazovek. Obsahuje navíc funkci, která umožňuje textový soubor se záznamem aktivit posílat pravidelně na danou e-mailovou adresu.[4]

2.1.5. Rescue Time

Jedná se o program zaznamenávající práci na počítači a z těchto záznamů počítá její efektivitu. Slouží především pro uživatele samotného. Prezentuje zaznamenaná data procentuálně podle toho, jak si je uživatel ohodnotil, například: Visual Studio – velmi pozitivní, čtení blogů – velmi negativní. Na základě tohoto rozdělení pak program počítá efektivitu. Program umožňuje zaznamenávat navštívené internetové adresy, spuštěné programy. Obsahuje také plánovač, omezovač (uživatel je upozorněn, že dané aktivitě věnoval již příliš mnoho času).[5]

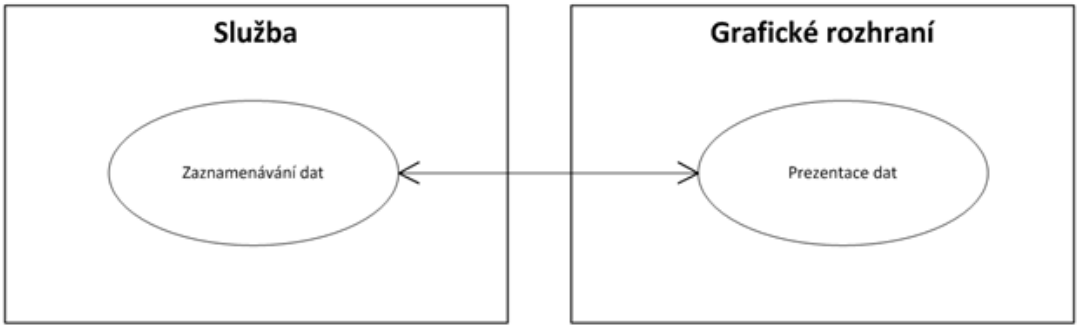


2.1.6. Tabulka vlastností aplikací

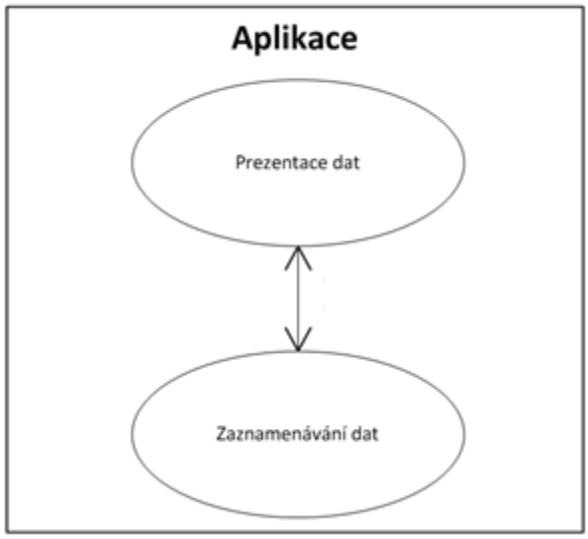
	Stisknuté klávesy	Adresy URL	Start/Stop mapování	Otisky obrazovek	Běžící aplikace	Určeno pro	Freeware
Revealer Keylogger	NE	ANO	ANO	NE	ANO	UŽIVATEL	ANO
User Logger	NE	ANO	ANO	ANO	ANO	UŽIVATEL	ANO
Activity Monitor	ANO	ANO	NE	ANO	ANO	ADMINISTRÁTOR	NE
Spytech SpyAgent	ANO	ANO	NE	ANO	ANO	ADMINISTRÁTOR	NE
Rescue Time	NE	ANO	ANO	NE	ANO	UŽIVATEL	ANO

3. Analýza

Sledování aktivit uživatelů – jedná se o rozsáhlý problém, jak mapovat a zaznamenávat činnosti, spuštěné aplikace uživatelů na daném počítači (a případně jejich vlastnosti). Existuje mnoho možností, jak tohoto mapování docílit. Nejzajímavější jsou z mého pohledu dvě. Jednou možností je vytvořit službu běžící na pozadí, která zaznamenává spouštěné aplikace a určitým způsobem je ukládá. K této službě vytvořit grafické rozhraní, které uložená data příhodně prezentuje. Toto řešení využívá můj program.



Jinou možností může být standardní aplikace, která po spuštění začne mapovat již běžící programy a v reálném čase tyto programy (a informace o nich) zobrazuje.



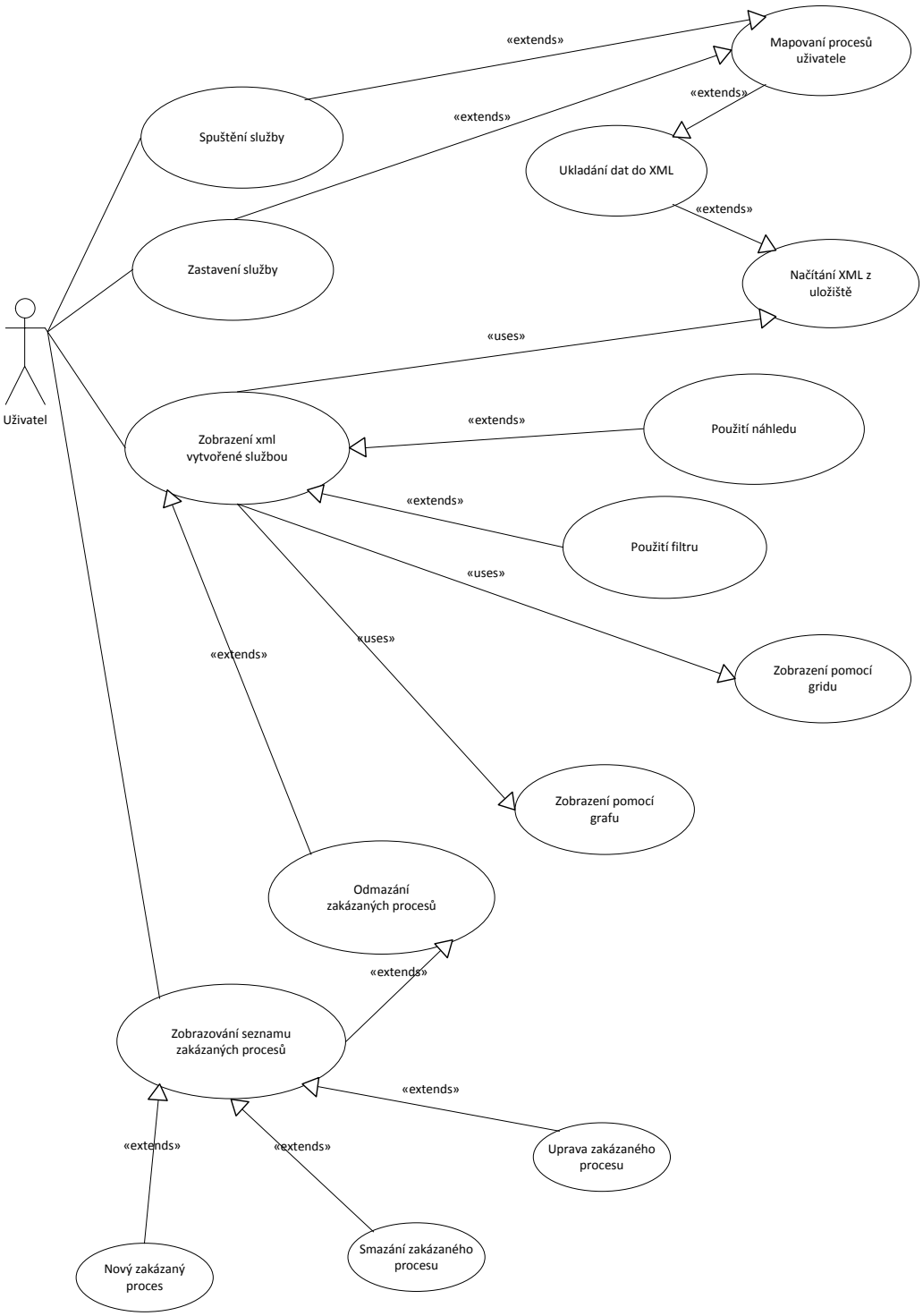
3.1. Moje řešení

Rozhodl jsem se použít koncept služby běžící na pozadí a desktopovou aplikaci pro prezentaci výsledků mapování činností. Pro tuto volbu jsem se rozhodl na základě zkušeností s již existujícími řešeními.

4. Návrh

Své řešení budu implementovat na platformě .NET Framework pomocí programovacího jazyka C#. Bude se tedy jednat o standardní službu Windows, pro desktopovou aplikaci si zvolím technologii Windows Presentation Foundation (WPF). Služba si bude v pravidelném intervalu zjišťovat běžící procesy a zaznamenávat je. Tato zaznamenaná data budou ukládána formou XML souboru do zvoleného úložiště. Aplikace WPF bude tyto data zpracovávat a prezentovat uživateli. Prezentace těchto dat bude formou grafu i výpisu. Bude také možné tyto data filtrovat a zobrazovat si pouze ty informace k datům, které uživatel chce. Další vlastností tohoto grafického rozhraní bude i možnost vytvářet si seznam dat, která se ve výpisu mají ignorovat například proto, že se jedná o systémové procesy, které uživatele nezajímají. Bude zde také možnost manuálně spouštět a zastavovat službu.

Use case


Uživatel
Spuštění služby
Zastavení služby
Mapovaní procesů uživatele
Ukladání dat do XML
Načítání XML z uložiště
Zobrazení xml vytvořené službou
Použití náhledu
Použití filtru
Zobrazení pomocí gridu
Zobrazení pomocí grafu
Odmazání zakázaných procesů
Zobrazování seznamu zakázaných procesů
Uprava zakázaného procesu
Smazání zakázaného procesu
Nový zakázaný proces
«extends»
«uses»

4.1. Použité technologie

4.1.1. Microsoft Windows služby

Jedná se o dlouho běžící spustitelné aplikace, které nevykazují žádné uživatelské rozhraní. Aplikace Windows služby běží pod Windows stanicí (Windows station), které není interaktivní, což má za následek, že se například dialogová okna volaná službou nebudou zobrazovat uživateli. Dokonce jejich volání v aplikaci Windows služby může způsobit, že aplikace služby přestane reagovat.

Proces Windows služby běží ve vlastním zabezpečeném kontextu a je spuštěn před přihlášením uživatele. Je nutné postupovat opatrně při přidělování možnosti ovládat službu přihlášeným uživatelem. Důvodem je, že služba běží pod systémovým účtem a má mnohem více privilegií než samotný uživatel.

Ve Windows prostředí je centrálním nástrojem pro správu služeb Windows ServiceControlManager, pomocí kterého lze manipulovat se službou.[6]

4.1.2. WPF – Windows Presentation Foundation

Jedná se o grafický prezentační systém společnosti Microsoft pro vytváření klientských aplikací Windows. Tato technologie rozšiřuje možnosti technologií ASP.NET a Windows Forms. WPF umožňuje používat značkovací jazyk XAML a zároveň kód v pozadí, přičemž vzhled je vytvářen pomocí značkovacího jazyka a chování je upravováno pomocí kódu v pozadí.[7]

5. Uživatelská příručka – „Watcher – Sledovací program“

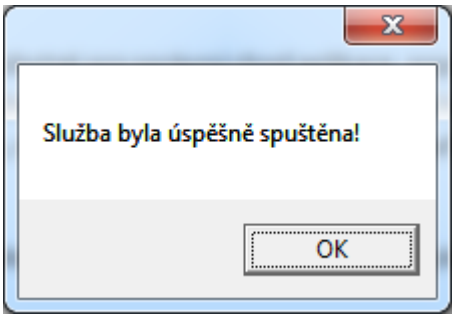
5.1. Instalace

Uživatel spustí instalační program „Watcher – Sledovací program.msi“. Zobrazí se Windows rozhraní pro instalaci aplikací „Průvodce instalací sady Watcher – Sledovací program“. V průběhu instalace si uživatel zvolí umístění, kam chce program Watcher nainstalovat.

5.2. První spuštění

Po úspěšném nainstalování je nezbytné pro správný chod aplikace poprvé spustit aplikaci jako administrátor. Důvodem je skutečnost, že při prvním spuštění dochází k zápisu do registrů, na což běžný uživatel nemá práva a aplikace by nefungovala korektně.

Při prvním spuštění se zobrazí informativní okno informující, že služba byla úspěšně spuštěna.



Pokud se zobrazí toto informativní okno, je aplikace plně připravena pro správnou funkčnost.

5.3. Rozložení aplikace

V horní části aplikace je možno přecházet mezi jednotlivými moduly.

Aplikace se skládá ze čtyř hlavních modulů:

Proces – Základní modul pro práci s procesy uživatele

Zakázané procesy – Modul zachycující procesy, které uživatel nechce zobrazovat

Nastavení – Modul pro nastavení vlastností aplikace i služby běžící na pozadí

O programu – Modul nesoucí informace o verzi produktu a o autorovi

5.3.1. Modul proces

Tento modul je nejdůležitějším pro uživatele. Právě zde se zobrazují informace o aktivitách uživatele, jejich filtrace, zobrazení a jejich následná prezentace či zálohování do externích souborů. Modul samotný je rozdělen na tři části. V levé části se nachází možnost filtrování a pohledů.

Ve filtru je možné vyplňovat výběrové podmínky, podle kterých se následně budou zobrazovat data. Vyhledávání je napsáno způsobem, že není nutné psát přesně hledaný výraz. Například, pokud uživatel vyhledává procesy pro výraz „ex“ budou v nalezených výsledcích i procesy „explore“ „iexplore“ a další.

Část pohled rozhoduje o zobrazených sloupcích ve výpisu informací. Pokud není označen ani jediný sloupec, potom to aplikace považuje, jako by byly vyplněny všechny.

Obě části filtr i pohled obsahují dvě tlačítka. První má dva stavy:

 - značí, že oblast bude při obnově brána v potaz

 - značí, že oblast nebude při obnově brána v potaz

Druhé tlačítko  slouží k rychlému vymazání hodnot v dané oblasti.

Posledním tlačítkem v této části je tlačítko obnovit  , které slouží pro aplikaci filtru a pohledu na zobrazovaná data.

V pravé horní části se zobrazuje panel nástrojů umožňující základní funkce se zobrazeným informacemi.

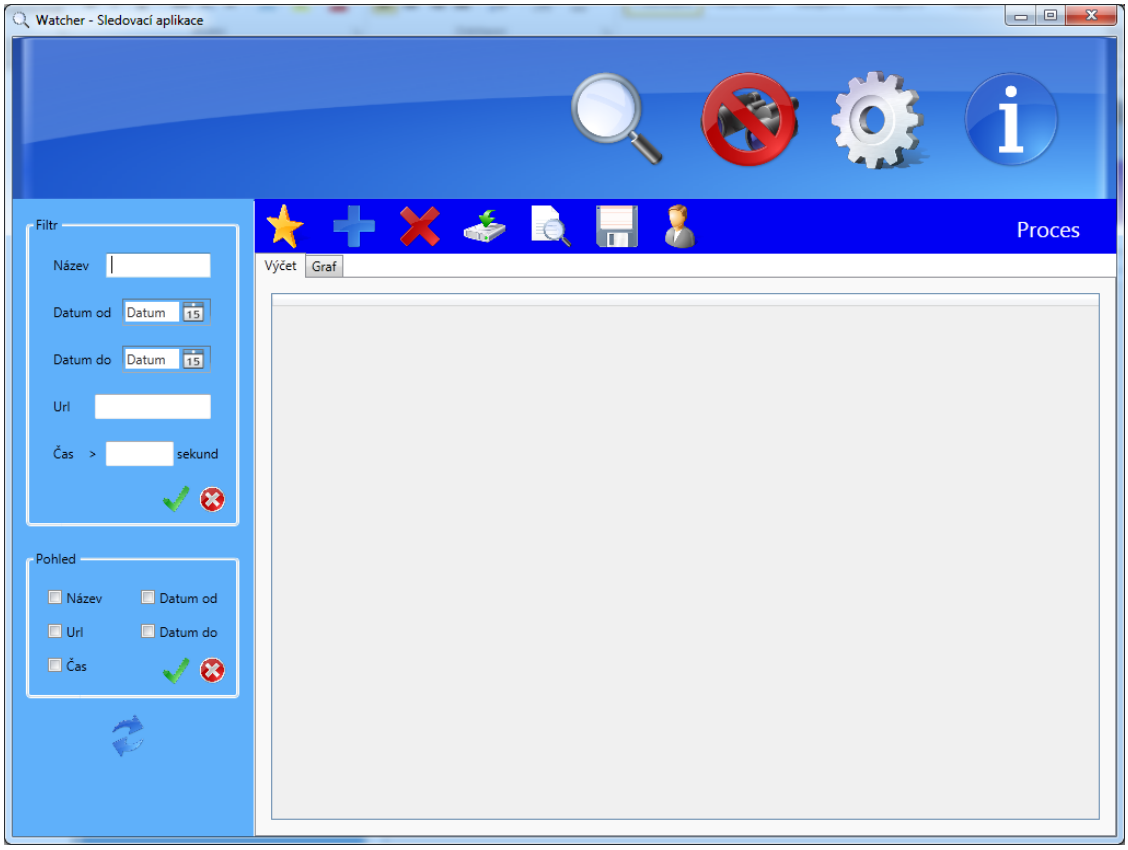
Jsou zde tyto volby:

 Načíst nové XML – tato volba slouží k načítání externích souborů XML, například od jiných uživatelů do programu. Při načítání budou předchozí výsledky smazány.

 Načíst a přidat XML – tato volba má stejnou funkci jako předchozí s tím rozdílem, že předchozí výsledky se nemažou, ale zůstávají v programu zobrazeny.

 Smazat vše – tato volba vymaže všechna zobrazená data

Zobrazení aplikace po použití volby „Smazat vše“



Uložit jako HTML – tato volba uloží zobrazená data do souboru HTML jako tabulku

Náhled – tato volba zobrazí náhled souboru HTML vytvořeného předchozí volbou

Příklad HTML náhledu

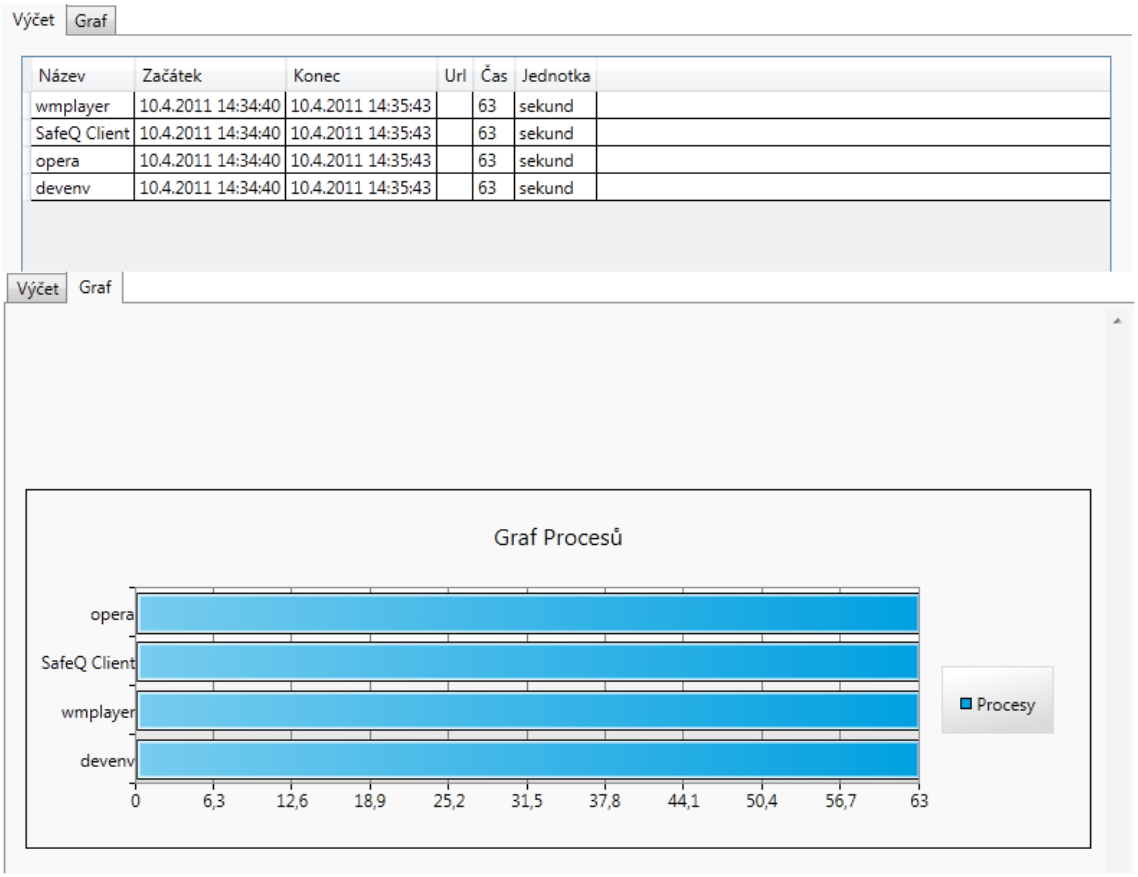
Název	Začátek	Konec	Url	Čas	Jednotka
dllhost	9.4.2011 20:40:48	9.4.2011 20:40:50		2	sekund
dllhost	9.4.2011 20:41:13	9.4.2011 20:41:20		7	sekund
WLIDSVCM	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund
VCDDaemon	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund
WDDMService	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund
WUDFHost	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund
QipGuard	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund
regedit	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund
WDSmartWareBackgroundService	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund
devenv	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund
realsched	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund
WLIDSVC	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund
StarWindServiceAE	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund
WDDMStatus	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund
WDSmartWare	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund
Idle	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund

Exportovat zobrazená data do XML – tato volba vytváří ze zobrazených dat XML soubor, který je následně možné načíst zpět do programu pomocí voleb „Načíst nové XML“ a „Načíst a přidat XML“

Obnovit procesy zachycené pro tohoto uživatele – tato volba zobrazí zachycené procesy pro aktuálního uživatele od prvního spuštění aplikace do tohoto momentu

Poslední částí je vpravo dole výpis zobrazovaných informací. Data mohou být vypsána dvěma způsoby: pomocí standardní tabulky nebo pomocí grafu.

Zobrazování procesů



5.3.2. Modul Zakázané procesy

Tento modul obsahuje procesy, které si uživatel nepřeje zobrazovat v modulu „Procesy“. V tomto modulu je možné tyto „Zakázané procesy“ vytvářet, editovat, filtrovat a mazat. Rozložení tohoto modulu je stejné jako u modulu Proces. I funkce jsou podobné, proto v této části budou popsány pouze ty funkce, které jsou odlišné od funkcí v modulu Proces.

V pravé horní části jsou nové volby:

Nový proces – tato volba zobrazí formulář pro vytvoření nového zakázaného procesu. Tento formulář obsahuje pouze pole pro vyplnění názvu a poznámky k procesu, dále možnosti Uložit a Zrušit. Po uložení se bude nový zakázaný proces zobrazovat ve výpisu.

Upravit – tato volba umožňuje upravit již existující zakázaný proces. Po vybrání této volby se zobrazí stejný formulář jako při vytváření nového zakázaného procesu. Po uložení se změny projeví ve výpisu.

V pravé dolní části je výpis zobrazován pouze formou standardní tabulky.

5.3.3. Modul Nastavení

V tomto modulu je uživateli umožněno pracovat s nastavením služby. Pro správnou činnost těchto funkcí je nezbytné, aby aplikace byla spuštěna s právy administrátora! Modul obsahuje tyto funkce:

 Spuštění Sledovací služby manuálně.

 Zastavení Sledovací služby manuálně.

Dále zde má uživatel možnost změnit frekvenci ukládání informací o vlastní aktivitě. Tato funkce se projeví až po restartování služby.

5.3.4. Modul O programu

Tento modul slouží pouze pro informativní účely. Zobrazuje autora aplikace a verzi aplikace.

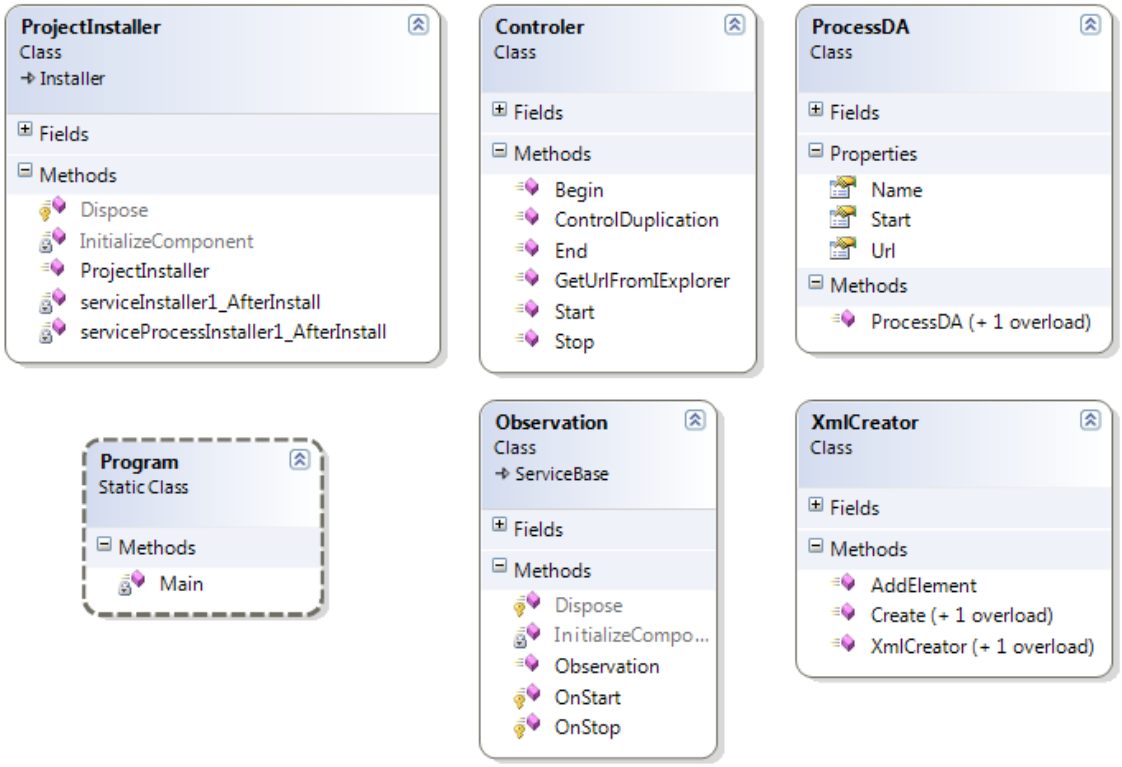
6. Implementace

Pro svou aplikaci jsem využil volně přístupné ikony a obrázky. [8]

6.1. Služba

Služba si opakovaně nechává vypsat běžící procesy a tvoří z nich objekty. Pokud se jedná o proces Internet Exploreru je u něho zjišťována i prohlížená URL adresa. Po ukončení procesu je tento objekt zapsán do XML dokumentu. Následně je z tohoto dokumentu vytvořen XML soubor. Ten se ukládá do umístění, které služba při spuštění zjišťuje z registrů. Stejně jako umístění se z registrů zjišťuje i frekvence, s jakou si služba nechává vypisovat běžící procesy. Služba je nastavena aby se automaticky spouštěla po spuštění počítače.

Výpis tříd



Třída *Observation*

Tato třída je základní třídou Windows služby. Obsahuje metody OnStart a OnStop, kde je definováno, co má služba provádět při spuštění a zastavení. Tato třída volá třídu Controler.

Třída *Controler*

V této třídě je popsána hlavní funkčnost služby. Činnosti metod:

- *Begin* – Zde se vytváří XML dokument pomocí třídy XmlCreator. Zaznamenávají se zde všechny právě probíhající procesy ve formě objektů ProcessDA.
- *ControlDuplication* – Tato metoda má za úkol odstranit duplicity. Je volána při každém zaznamenávání právě probíhajících procesů.
- *GetUrlFromIExplorer* – Pomocí této metody se z procesu Internet Exploreru získávají otevřené URL adresy v prohlížeči. Toto ULR je společně s dalšími informacemi o procesu ukládána do objektu ProcessDA. Tato metoda je volána vždy, když program zjistí, že se jedná o proces Internet Exploreru.[9]

```
        SHDocVw.ShellWindows shellWindows = new SHDocVw.ShellWindows();

                    string filename;

                    foreach (SHDocVw.InternetExplorer ie in shellWindows)
                    {
                        filename                                                    =
Path.GetFileNameWithoutExtension(ie.FullName).ToLower();

                        if (filename.Equals("iexplore"))
                        {
                            urls = ie.LocationURL;
                        }
                    }
```

- *End* – Zde se všechny objekty ProcessDA přidají do XML dokumentu třídy XmlCreator a pomocí této třídy se XML dokument uloží na disk.
- *Start* – V této metodě je popsána největší část funkčnosti služby. V první části se zde z registrů zjišťují potřebné parametry frekvence ukládání běžících procesů a umístění, kam se následně bude ukládat XML soubor s daty. Metoda dále obsahuje nekonečný cyklus, který zjišťuje běžící procesy, zaznamenává je do objektů ProcessDA a po ukončení jejich běhu je ukládá do XML dokumentu. V závislosti na frekvenci pak z XML dokumentu vytváří XML soubor.
- *Stop* – Tato metoda volá metodu End.

Třída *XmlCreator*

Tato třída zajišťuje práci s XML dokumentem a stará se o vytváření a ukládání XML souboru.

- *XmlCreator* – Konstruktor, ve kterém je vytvořen XML dokument.
- *AddElement* – Metoda slouží pro přidávání elementů do XML dokumentu.
- *Create* – Tato metoda ukládá XML soubor.

Třída *ProcessDA*

Je datová třída sloužící pro ukládání žádoucích informací o procesu. V konstruktoru mohou být splněny všechny vlastnosti. Vlastnosti jsou veřejné a mohou být upravovány i postupně.

Třída *ProjectInstaller*

V této třídě je nastavována správná instalace služby Windows, způsob spouštění služby (automatické) a pod jakým účtem bude služba nainstalována.

Struktura ukládaného XML dokumentu

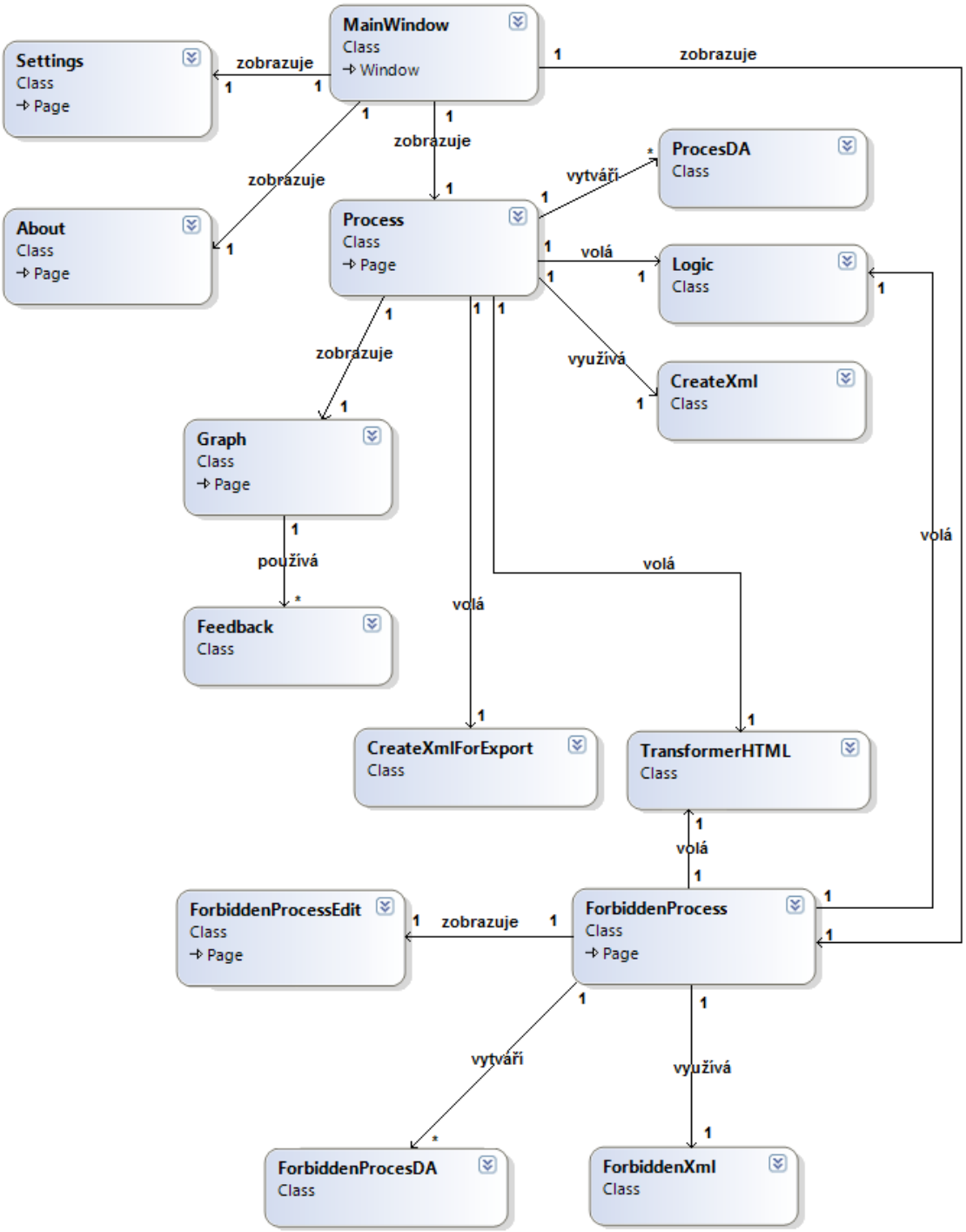
```
<?xml version="1.0" encoding="utf-8" ?>
- <xs:schema id="Log" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  - <xs:element name="Log" msdata:IsDataSet="true" msdata:Locale="en-US">
    - <xs:complexType>
      - <xs:choice minOccurs="0" maxOccurs="unbounded">
        - <xs:element name="proces">
          - <xs:complexType>
            - <xs:sequence>
              - <xs:element name="url" minOccurs="0" maxOccurs="unbounded">
                - <xs:complexType>
                    <xs:attribute name="value" type="xs:string" />
                  </xs:complexType>
                </xs:element>
              </xs:sequence>
              <xs:attribute name="name" type="xs:string" />
              <xs:attribute name="start" type="xs:string" />
              <xs:attribute name="stop" type="xs:string" />
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
```

Příklad konkrétního XML dokumentu

```
<?xml version="1.0" ?>
- <Log>
  - <proces name="dllhost" start="12.4.2011 16:41:41" stop="12.4.2011 16:41:46">
      <url value="" />
    </proces>
  - <proces name="wininit" start="12.4.2011 16:41:41" stop="12.4.2011 16:44:42">
      <url value="" />
    </proces>
  - <proces name="svchost" start="12.4.2011 16:41:41" stop="12.4.2011 16:44:42">
      <url value="" />
    </proces>
  - <proces name="ViewAplication_WPF_2.0.vshost" start="12.4.2011 16:41:41" stop="12.4.2011 16:44:42">
      <url value="" />
    </proces>
  - <proces name="WDDMService" start="12.4.2011 16:41:41" stop="12.4.2011 16:44:42">
      <url value="" />
    </proces>
  - <proces name="dwm" start="12.4.2011 16:41:41" stop="12.4.2011 16:44:42">
      <url value="" />
    </proces>
  </Log>
```

6.2. Aplikace

Celkový diagram tříd



Aplikace při prvním startu zapisuje do registrů umístění, kam byla nainstalována a kam se má ukládat XML soubor vytvářený službou. Taktéž zapisuje do registru frekvenci, s jakou si služba nechává vypisovat běžící procesy.

```
        if (ConfigurationManager.AppSettings["firstStart"] == "0")
        {
            //zapis cesty do registru
            Microsoft.Win32.RegistryKey key;
            key                                                    =
Microsoft.Win32.Registry.LocalMachine.CreateSubKey(@"Software\Watcher");
            key.SetValue("Way", path);
            key.Close();

            Microsoft.Win32.RegistryKey key2;
            key2                                                   =
Microsoft.Win32.Registry.LocalMachine.CreateSubKey(@"Software\Watcher");
            key2.SetValue("Timer", "180");
            key2.Close();

            //prepis configu
            System.Configuration.Configuration            config           =
ConfigurationManager.OpenExeConfiguration(ConfigurationUserLevel.None);
             config.AppSettings.Settings["firstStart "].Value = "1";
             config.Save(ConfigurationSaveMode.Modified);
             ConfigurationManager.RefreshSection("firstStart ");
        }
```

Při implementaci jsem se soustředil na hlavní funkce aplikace:

- načítání a prezentace dat z XML souboru vytvořeného službou
- možnost exportovat zobrazovaná data
- možnost importovat na jiném počítači exportovaná data

Vedlejší funkce aplikace jsou:

- vytváření seznamu zakázaných procesů
- ovládání služby (spuštění, zastavení, změna frekvence ukládání)
- export dat do HTML

Příklad struktury ukládaného HTML dokumentu

```
        <html>
        <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Výčet procesů</title>
        </head>
        <body>
        <table border="5"
        <tr><td>Název</td><td>Začátek</td><td>Konec</td><td>Url</td><td>Čas<
/td><td>Jednotka</td></tr>
        <tr><td>SearchProtocolHost</td><td>9.4.2011
20:40:48</td><td>9.4.2011
20:41:17</td><td></td><td>29</td><td>sekund</td></tr>
```

```
      <tr><td>SearchFilterHost</td><td>9.4.2011  20:40:48</td><td>9.4.2011
20:41:17</td><td></td><td>29</td><td>sekund</td></tr>
      <tr><td>sqlwriter</td><td>9.4.2011         20:40:48</td><td>9.4.2011
20:41:25</td><td></td><td>37</td><td>sekund</td></tr>
      <tr><td>sqlservr</td><td>9.4.2011          20:40:48</td><td>9.4.2011
20:41:25</td><td></td><td>37</td><td>sekund</td></tr>
      <tr><td>VCDDaemon</td><td>9.4.2011         20:40:48</td><td>9.4.2011
20:41:25</td><td></td><td>37</td><td>sekund</td></tr>
      <tr><td>WDDMService</td><td>9.4.2011       20:40:48</td><td>9.4.2011
20:41:25</td><td></td><td>37</td><td>sekund</td></tr>
      <tr><td>soffice</td><td>9.4.2011           20:40:48</td><td>9.4.2011
20:41:25</td><td></td><td>37</td><td>sekund</td></tr>
      <tr><td>regedit</td><td>9.4.2011           20:40:48</td><td>9.4.2011
20:41:25</td><td></td><td>37</td><td>sekund</td></tr>
      <tr><td>opera</td><td>9.4.2011             20:40:48</td><td>9.4.2011
20:41:25</td><td></td><td>37</td><td>sekund</td></tr>
      <tr><td>omtsreco</td><td>9.4.2011          20:40:48</td><td>9.4.2011
20:41:25</td><td></td><td>37</td><td>sekund</td></tr>
      <tr><td>SafeQ    Client</td><td>9.4.2011    20:40:48</td><td>9.4.2011
20:41:25</td><td></td><td>37</td><td>sekund</td></tr>
      <tr><td>soffice.bin</td><td>9.4.2011       20:40:48</td><td>9.4.2011
20:41:25</td><td></td><td>37</td><td>sekund</td></tr>
      <tr><td>devenv</td><td>9.4.2011            20:40:48</td><td>9.4.2011
20:41:25</td><td></td><td>37</td><td>sekund</td></tr>
      <tr><td>Observation_V1</td><td>9.4.2011    20:40:48</td><td>9.4.2011
20:41:25</td><td></td><td>37</td><td>sekund</td></tr>
      <tr><td>realsched</td><td>9.4.2011         20:40:48</td><td>9.4.2011
20:41:25</td><td></td><td>37</td><td>sekund</td></tr>
      <tr><td>StarWindServiceAE</td><td>9.4.2011 20:40:48</td><td>9.4.2011
20:41:25</td><td></td><td>37</td><td>sekund</td></tr>
      <tr><td>WDSmartWare</td><td>9.4.2011       20:40:48</td><td>9.4.2011
20:41:25</td><td></td><td>37</td><td>sekund</td></tr>
      <tr><td>Idle</td><td>9.4.2011              20:40:48</td><td>9.4.2011
20:41:25</td><td></td><td>37</td><td>sekund</td></tr>
      <tr><td>iexplore</td><td>9.4.2011          20:40:52</td><td>9.4.2011
20:41:25</td><td></td><td>33</td><td>sekund</td></tr>
      </table>
      </body>
      </html>
```

Příklad HTML souboru

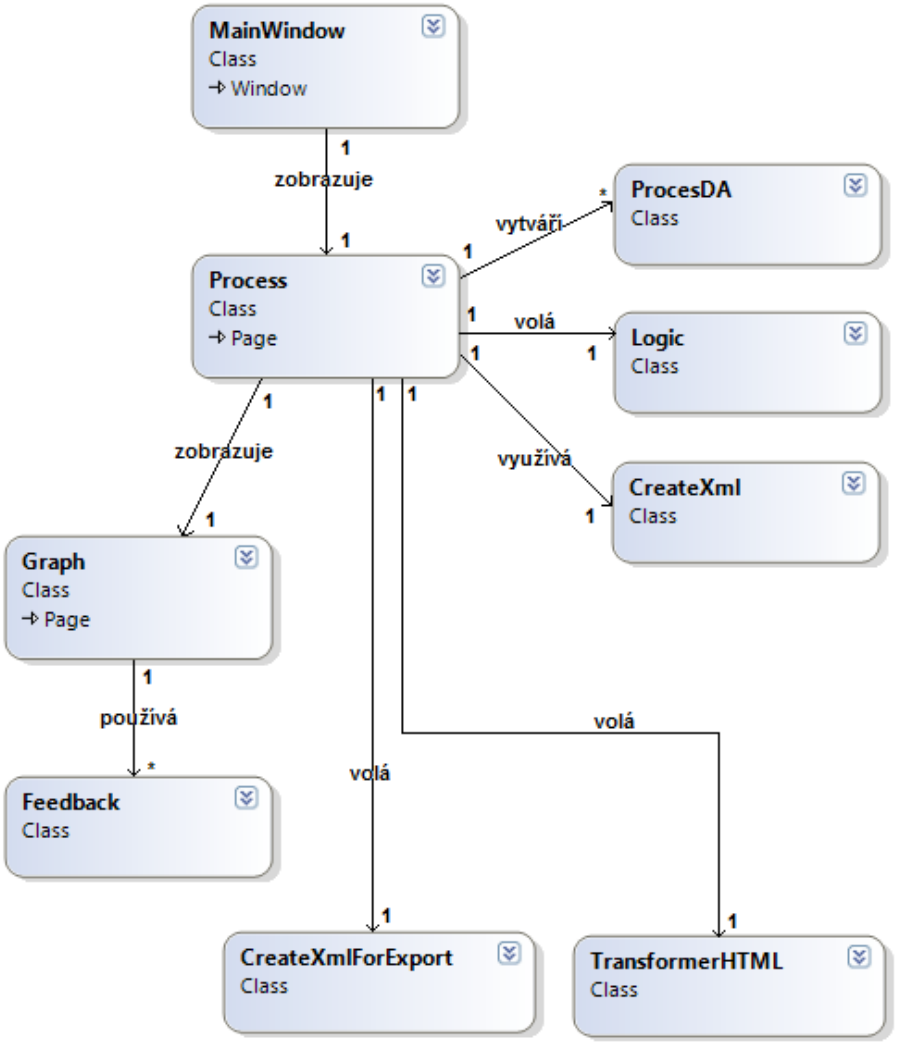
Název	Začátek	Konec	Url	Čas	Jednotka
dllhost	9.4.2011 20:40:48	9.4.2011 20:40:50		2	sekund
dllhost	9.4.2011 20:41:13	9.4.2011 20:41:20		7	sekund
WLIDSVCM	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund
VCDDaemon	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund
WDDMService	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund
WUDFHost	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund
QipGuard	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund
regedit	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund
WDSmartWareBackgroundService	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund
devenv	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund
realsched	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund
WLIDSVC	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund
StarWindServiceAE	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund
WDDMStatus	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund
WDSmartWare	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund
Idle	9.4.2011 20:40:48	9.4.2011 20:41:25		37	sekund

Aplikace obsahuje hlavní okno, které využívá čtyři moduly – Process, ForbiddenProcess, Settings a About.

6.2.1.Process

Mezi funkce této části patří načítání zdrojového XML souboru, zobrazování načtených dat, filtrace, pohledy, export do HMTL, export do XML. Zobrazovaná data k procesům jsou název, začátek, konec, url, čas a jednotka.

Diagram tříd



Třída *Process* zaštiťuje všechny funkčnosti tohoto modulu. K načítání dat z XML souboru využívá třídu *Logic*. V této třídě jsou data překopírována do Listu objektů *ProcesDA*. Tyto objekty jsou následně zobrazovány pomocí výpisu ve třídě *Process* nebo pomocí grafu ve třídě *Graph*. Filtrování a možnost zobrazování pouze některých sloupců informací se provádí přímo ve třídě *Process*. Další metody a funkčnosti:

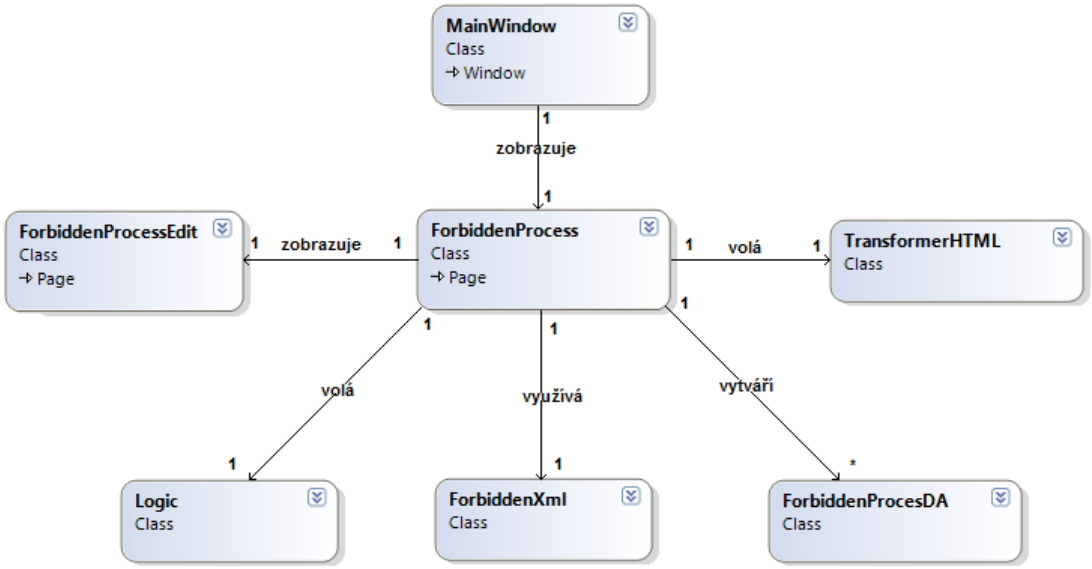
- Metoda *Synchronize* – tato metoda zajišťuje spojování dvou i více procesů v případě, že na sebe navazují bez přestávky, nebo se svými časovými obdobími překrývají.
- Metoda *StarService* – pomocí této metody je manuálně spouštěna služba mapující běžící procesy. Tato metoda je z této třídy volána pouze při prvním spuštění.
- Metoda *CompareURL* – zde se v případě spojování procesů pomocí metody *Synchronize* provádí odstraňování duplicit u zaznamenaných URL adres

- Metoda *CreateXMLfromGrid* – zajišťuje export právě zobrazovaných dat (filtrovaných) do XML souboru se stejnou strukturou, jakou má zdrojový XML soubor. K tomuto záměru využívá třídu *CreateXmlForExport*
- Metoda *DeleteForbiddenProcesses* – se stará, aby se ve výpisu procesů nezobrazovaly ty procesy, které si uživatel dlouhodobě nepřeje zobrazovat. Tyto procesy jsou zaznamenány v druhém modulu ForbiddenProcess (popsáno níže).
- Metoda *MyRefresh* – zajišťuje při aktualizaci výpisu dat, že se provedou všechny dílčí kroky pro správný výpis. Mezi tyto kroky mimo jiné patří: volání metody *DeleteForbiddenProcesses*, volání metody *Synchronize*, aplikace filtrů, aplikace pohledů (zobrazování pouze uživatelem zvolených sloupců ve výpisu) a další
- Třída *TransformerHTML* – se stará o vytváření HTML souborů ze zdrojových dat

6.2.2. ForbiddenProcess

Mezi funkce tohoto modulu patří zobrazování zakázaných procesů (procesy, které si uživatel dlouhodobě nepřeje zobrazovat ve výpisu procesů; data jsou uložena v XML souboru), vytváření, editace, mazání a filtrace těchto procesů, export do HTML souboru. Zobrazovaná data k zakázaným procesům jsou název, vytvořeno, změněno a poznámka.

Diagram tříd



Třída *ForbiddenProcess* zaštiťuje všechny funkce tohoto modulu. K načítání dat z XML souboru využívá třídu *Logic*. V této třídě jsou data překopírována do Listu objektů *ForbiddenProcesDA*. Tyto objekty jsou následně zobrazovány pomocí datagridu. Filtrování a možnost zobrazování pouze některých sloupců informací se provádí přímo ve třídě

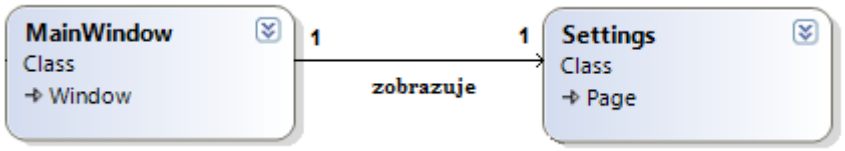
ForbiddenProcess. K vytváření nových a editaci již existujících zakázaných procesů je využívána třída *ForbiddenProcessEdit*. Další metody:

- Třída *TransformerHTML* – se stará o vytváření HTML souborů ze zdrojových dat

6.2.3.Settings

Mezi funkce tohoto modulu patří manipulace se službou, spuštění, zastavení a změna frekvence ukládání XML souborů ve službě.

Diagram tříd



Třída Settings zaštiťuje všechny funkčnosti tohoto modulu.

Spouštění služby

```
        ServiceController service = new ServiceController(serviceName);
                    try
                    {
                        TimeSpan                    timeout                    =
TimeSpan.FromMilliseconds(timeoutMilliseconds);
                        service.Start();

service.WaitForStatus(ServiceControllerStatus.Running, timeout);
                        MainWindow.ExceptionCall("Služba     byla     úspěšně
spuštěna!");
                    }
                    catch (Exception e)
                    {
                        MainWindow.ExceptionCall("Služba    nebyla    spuštěna
kvůli vnitřní chybě!", e.StackTrace);
```

Zastavování služby

```
        ServiceController service = new ServiceController(serviceName);
                    try
                    {
                        TimeSpan                    timeout                    =
TimeSpan.FromMilliseconds(timeoutMilliseconds);
                        service.Stop();

service.WaitForStatus(ServiceControllerStatus.Stopped, timeout);
                        MainWindow.ExceptionCall("Služba     byla     úspěšně
zastavena!");
```

```
                    }
                    catch (Exception e4)
                    {
                        MainWindow.ExceptionCall("Služba nebyla zastavena
kvůli vnitřní chybě!",e4.StackTrace);
                    }
```

Změna frekvence ukládání XML souboru službou

```
        try
                        {
                            Microsoft.Win32.RegistryKey key2;
                            key2                                        =
Microsoft.Win32.Registry.LocalMachine.CreateSubKey(@"Software\Watcher");
                            key2.SetValue("Timer", textBox_novy_pocet.Text);
                            key2.Close();
                            Labe_pocet_vterin.Content                   =
textBox_novy_pocet.Text;

                            MainWindow.ExceptionCall("Frekvence     ukládání
byla úspěšně změněna.");
                        }
                        catch (Exception e2)
                        {
                            MainWindow.ExceptionCall("Nedošlo    ke    změně
frekvence z důvodu vnitřní chyby.", e2.StackTrace);
                        }
```

7. Zhodnocení

Považuji svůj program za povedený. Program by bylo možné rozšířit o ukládání dat do externí databáze, přidání nových funkcí do uživatelského rozhraní, jako například poměry časů pracovních programů s mimopracovními programy. Dalším možným vylepšením je pro programování v technologii WPF použití návrhového vzoru MVVM (Model-View-ViewModel) pro lepší oddělení aplikační logiky od prezentační složky (tato vlastnost by mohla být užitečná při aktualizaci nových verzí). Pro lepší ovládání služby, by bylo možné doplnit do oznamovací oblasti hlavního panelu ikonu s funkcemi: zastav, spusť, pozastav na x minut, restartuj, atd.

8. Závěr

Při psaní této práce jsem se blíže seznámil s technologií Windows services a Windows Presentation Foundation. Prohloubil jsem si své schopnosti v programování na platformě .NET Framework programovacím jazyku C#. Získal jsem povědomí, jakým způsobem lze zachytávat a mapovat procesy, které jsou spouštěny v systému Windows. Vyzkoušel jsem si práci s registry i s možností, jak získávat navštívené URL adresy z internetového prohlížeče Internet Explorer.

9. Použitá literatura

1. *Revealer Keylogger (Free Edition) download and reviews from SnapFiles.com* [online]. c2009, poslední revize 2.4.2009 [cit. 2011-03-26]. Dostupné z: <http://www.snapfiles.com/get/revealerfree.html>
2. *User Logger download and reviews from SnapFiles.com* [online]. c2003, poslední revize 26.4.2003 [cit. 2011-03-26]. Dostupné z: <http://www.snapfiles.com/get/usrlogger.html>
3. *Activity Monitor download and reviews from SnapFiles.com* [online]. c2010, poslední revize 11.1.2011 [cit. 2011-03-26]. Dostupné z: <http://www.snapfiles.com/get/activitymonitor.html>
4. *Spytech SpyAgent download and reviews from SnapFiles.com* [online]. c2010, poslední revize 18.3.2011 [cit. 2011-03-26]. Dostupné z: <http://www.snapfiles.com/get/spyagent.html>
5. *Time Management, Productivity, & Project Tracking Software(Mac/PC)| Rescue Time* [online]. c2011 [cit. 2011-03-26]. Dostupné z: < http://www.rescuetime.com/>
6. *Introduction to Windows Service Applications* [online]. c2010, poslední revize 10.10.2010 [cit. 2011-03-26]. Dostupné z: <http://msdn.microsoft.com/en-us/library/d56de412(v=vs.80).aspxl>
7. *Introduction to WPF* [online]. c2011 [cit. 2011-03-26]. Dostupné z: <http://msdn.microsoft.com/en-us/library/aa970268.aspx>
8. *free Must Have Icon Set :: By VisualPharm :: Free For Commercial Use :: Available in png Format* [online]. c2011 [cit. 2010-06-16]. Dostupné z: <http://www.gettyicons.com/free-icon/112/must-have-icon-set/>
9. *Access Explorer and Internet Explorer in C# to Find Web Pages and Directories >> OmegaMan's Musings* [online]. c2007, poslední revize 21.6.2007 [cit. 2010-06-16]. Dostupné z: *<http://omegacoder.com/?p=63>*